Sequence-aware Heterogeneous Graph Neural Collaborative Filtering

Chen Li * Linmei Hu * Chuan Shi *†‡ Guojie Song § Yuanfu Lu ¶

Abstract

With the booming of the internet, a popular recommendation scenario has played a vital role in information acquisition for user where the latent heterogeneous collaborative signals and sequential patterns underlying a user's historical behaviors are important for better inferring which item she prefers to interact with next time. Traditional heterogeneous information network based methods or sequential recommendation methods either consider only heterogeneous collaborative signals in the interactions or model user embedding based on only their own item interaction sequence, which either can hardly capture a user's dynamic preferences or face a common data sparsity problem. In this paper, we propose a novel Sequence-aware Heterogeneous graph neural Collaborative Filtering model, called SHCF, which can address the above problems by considering both the high-order heterogeneous collaborative signals and sequential information. Specifically, we first construct a heterogeneous information network (HIN) by enriching the user-item bipartite graph with additional attribute information, and then design novel message passing layers for learning user and item embedding. For user embedding, we consider the sequential information to capture user's dynamic interests over time with a position-aware self-attention mechanism, and capture user's fine-grained static preferences on different aspects of an item with an element-wise attention mechanism. For item embedding, we carefully incorporate the heterogeneous attribute information with dual-level attention, which alleviates the data sparsity problem. Extensive experiments on three real-world datasets illustrate that our model can improve the recommendation performance compared with the state-of-the-art methods.

1 Introduction

In the era of information explosion, information overload has become a challenge for people to find the information that they are interested in. The recommendation system as an information filtering system that can learn the user's interests and hobbies based on the user's profile or historical behavior records, is widely applied in many Web applications, e.g., e-commerce, search, and media streaming sites. Based on the fact that users with similar preferences may be interested in similar items, most of the existing recommendation methods mine such collaborative information through user-item interactions, known as collaborative filtering (CF). Matrix factorization (MF) [11] is one of the most successful methods among CF techniques, which models the user preference as the inner product of user and item latent representations.

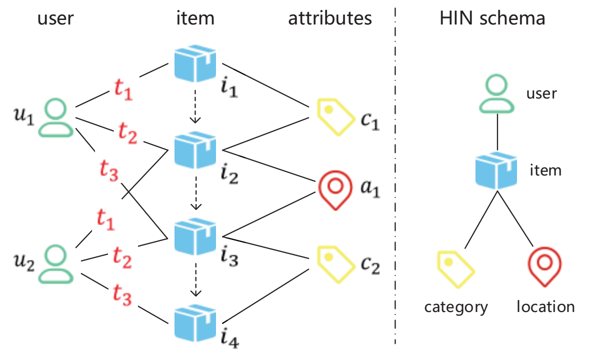


Figure 1: A toy example of user-item interactions with heterogeneous information and sequential patterns.

However, collaborative filtering methods suffer from the data sparsity problem, also called cold-start problem [19], thus many researchers consider blending more side information for hybrid recommendation. Heterogeneous information network (HIN) as an effective information fusion method containing different types of nodes and links, can be used to integrate multiple types of objects and their complex interactions in the recommendation system that may produce more accurate recommendation results [18]. Initially, many works leverage meta-path based semantic relatedness between users and items over HIN for recommendation [28, 4]. More recently, some works adopt network embedding models to learn latent user and item representations for rating prediction [17]. Despite their effectiveness, we argue that these methods consider only the heterogeneous collaborative information to model a user's static prefer-

*Beijing University of Posts and Telecommunications, China. Email: {leechen, hulinmei, shichuan}@bupt.edu.cn
†Peng Cheng Laboratory, China
‡Corresponding author
§Peking University, China. Email: gjsong@pku.edu.cn
¶WeChat Search Application Department, Tencent, China. Email: luyfroot@gmail.com

ence with the assumption that all user-item interactions in the historical sequence are equally important, while ignoring the interactions' sequential pattern that a small set of the most recent interactions can better reflect a user's dynamic interests over time.

Considering the sequential pattern to model a user's latest interests, there is another line of work called sequential recommendation. The sequential recommendation system is to predict which item a user most probably would like to interact with next time given her sequential interaction data as context. Markov Chain (MC) is a classic example for sequential recommendation, which assumes that the next action is conditioned on only the previous action [16]. Recently, many works use Recurrent Neural Networks (RNNs) to summarize all previous actions with a hidden state, which is used to predict the next action [6, 12, 7, 14]. However, almost all of the sequential recommendation methods model user embeddings based on only their own sequential item interactions while ignoring the heterogeneous information widely existing in recommendation system, such like item attributes. When the data is sparse and there are few user interaction behaviors, these methods also suffer from cold-start problem.

Based on these observations, it motivates us to conduct sequential recommendation on HIN settings, since these scenarios are very popular in real applications. For example, in e-commerce, there are multiple-type objects (e.g. user, item, agent, category, etc.) and multiple-type relationships (e.g. buy, click, belong, etc.), the platform can predict which item a user may interact with next time based on her interaction sequence. Moreover, it is potential to promote the recommendation performance when considering both sequential patterns and heterogeneous information. Taking Fig. 1 for example, the user u_1 interacts with items $\{i_1, i_2, i_3\}$ in the time order $\{t_1, t_2, t_3\}$, and the user u_2 interacts with i_4 after i_3, these item sequential dependencies can imply u_1's latest interests. Besides, the added item attributes such as c_j and a_j can bring richer semantic information to alleviate the data sparsity problem.

In this paper, we propose a novel **S**equence-aware **H**eterogeneous graph neural **C**ollaborative **F**iltering model (SHCF) to fully consider both the sequential patterns and the high-order heterogeneous collaborative signals. More specifically, we designed a novel heterogeneous graph neural network (GNN) for learning user and item representations in the HIN, which can capture heterogeneous collaborative information as well as incorporate sequential information during message propagation. For user embedding, we aggregate the user's interacted item embeddings with a novel element-wise attention mechanism, which assumes each dimension of the item embedding reflects a distinct aspect of the item and a user may prefer different aspects of the item. We also consider a user's dynamic interests by aggregating her interacted item sequence with a sequence-aware self-attention mechanism, where each item is correlated with a position embedding and self-attention is used to pay attention to important items reflecting her latest interests. For item embedding, we aggregate the heterogeneous information of its neighboring nodes including users and item attributes with dual-level attention. In this way, we can not only learn the importance of different nodes but also pay attention to important types of nodes. By stacking multiple message passing layers, we can enforce the embeddings to capture the high-order collaborative relationships. Experimental results on real-world datasets show that our model significantly outperforms state-of-the-art methods.

We summarize the contributions as follow:

- To our best knowledge, this is the first attempt to fully consider both the sequential patterns and the high-order heterogeneous collaborative signals in recommendation system to improve the performance of recommendation.

- We propose a novel sequence-aware heterogeneous graph neural collaborative filtering model SHCF, which incorporates additional attribute information for enriching user and item embedding. In addition, our model captures a user's dynamic interests over time with a sequence-aware self-attention mechanism.

- We conduct extensive experiments on real-world datasets to evaluate the performance of the proposed model. The results show the superiority of our model over the state-of-the-art models.

2 Related Work

2.1 Collaborative Filtering Recommendation

Collaborative filtering is a recommendation technique that can filter out items that a user might like based on reactions by similar users. Matrix factorization (MF) is one of the most popular collaborative filtering methods and has shown its effectiveness and efficiency in many applications [11, 10, 15]. MF factorizes the rating matrix into two low-rank user-specific and item-specific latent representations and then applies an inner product on them for prediction. Recently, with the development of deep learning techniques, neural CF models appear. Instead of modeling the user preference on an item as the inner product like traditional MF methods, He et al. proposed NeuMF [5] to leverage a multi-layer perceptron to learn the user-item interaction function with non-linearities. DMF [26] uses multiple nonlinear layers to process both explicit ratings and implicit feedbacks of users and items. Another line of research exploits the user-item interaction graph to infer user preference[27, 1, 22]. Yang et al. proposed HOP-Rec [27] which performs random walks to enrich the interactions of a user with multi-hop connected items. GC-MC [1] uses two multi-link graph convolution layers to aggregate user and item features. NGCF [22] explicitly incorporates collaborative signals by leveraging high-order connectivities in the user-item interaction graph.

2.2 Heterogeneous Information Network based Recommendation Facing the common data sparsity problem in recommendation tasks, many researchers consider blending more side information such as social relations [24, 30], knowledge graph [21], and heterogeneous information network (HIN) [29] for hybrid recommendation. HIN as one of the most important information fusing frameworks can naturally characterize different relations between different types of objects, thus it has attracted much attention. Feng et al. [3] proposed an optimization-based graph method for personalized tag recommendation which incorporates different sources of information to alleviate the cold start problem. Yu et al. [28] proposed to diffuse user preferences along different meta-paths in information networks. Shi et al. [17] designed a new random walk strategy based on meta-paths to derive more meaningful node sequences for node embedding and then integrated the embeddings into an extended matrix factorization model for recommendation. Han et al. [4] proposed NeuACF which explores aspect-level information extracted from heterogeneous network with meta-paths for collaborative filtering. Despite their effectiveness, all these methods do not consider the interactions' sequential pattern.

2.3 Sequential Recommendation Another line of work is the sequential recommendation which is to predict which item a user most probably would like to interact with next time given her sequential interaction data as context. FPMC [16] is a traditional method for the sequential recommendation. It utilizes MF and Markov chains to capture the long-range and short-range item transitions respectively. Other than MC-based methods, some works adopt RNNs to model user's interaction sequences [6, 12, 7, 14]. GRU4Rec [6] uses Gated Recurrent Units (GRU) to model click sequences for session-based recommendation. NARM [12] employs RNN with attention mechanism to capture users' features of sequential behavior and main purposes. Recently, attention mechanisms have been incorporated into recommender systems [13, 9, 25]. For example, STAMP [13] applies an attention net to capture both users' current interests and general interests. SASRec [9] models the entire interaction sequence with self-attention for the next item recommendation. SR-GNN [25] presents a novel architecture for the session-based recommendation that first models session sequences as graph structure data and develops an attention strategy to combine long-term interests and current interests to better predict users' next actions. Although these methods achieve promising performances in real applications, they model user embedding by just considering their own sequential item interactions while ignoring the heterogeneous information.

Different from the existing works, we propose a novel sequence-aware heterogeneous graph neural network, which takes full advantage of both the sequential interaction pattern and the high-order heterogeneous collaborative signals in recommendation system to improve the performance of recommendation.

3 Methodology

In this section, we present our proposed **S**equence-aware **H**eterogeneous graph neural **C**ollaborative **F**iltering (SHCF) model for recommendation, which takes full advantage of both sequential information and heterogeneous collaborative information. Fig. 2 shows the framework of our proposed model. Our model contains three steps. First, we construct an HIN with user-item interactions and item attributes as shown in Fig. 1. Note that here we only consider item attributes in order to focus on clearly illustrating how to handle sequence patterns and heterogeneous information. In fact, user attributes and other heterogeneous information can be easily added into our SHCF through concatenating embedding learned from these heterogeneous attributes as item embedding does. Then we apply an embedding layer to initialize the representations of users, items, and item attributes (e.g., item categories). Second, we design multiple message passing layers over the HIN to learn the user and item embeddings. For user embedding, we capture a user's fine-grained static interests on different aspects of an item with an element-wise attention mechanism. We also consider a user's dynamic interests by aggregating her interacted item sequence with a sequence-aware self-attention mechanism. For item embedding, we aggregate the heterogeneous information of its neighboring nodes including users and item attributes with dual-level attention which considers the importance of different neighboring nodes with different types. Finally, the prediction layer aggregates the learned embeddings from different message passing layers for both user and item representations, and outputs

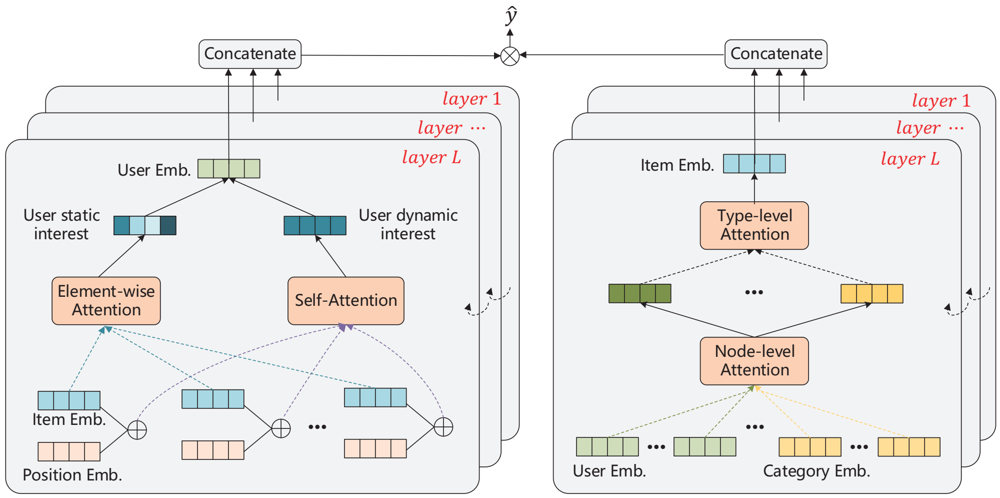


Figure 2: The framework of SHCF.

the prediction score of the target user-item pair.

3.1 Embedding Layer The users, items and item attributes in real datasets are usually identified by some unique IDs, whereas these original IDs have a very limited representation capacity. Therefore, we create a user embedding matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{U}| \times d}$, where d is dimensionality of the latent embedding spaces, and the j^{th} row of the embedding matrix $\mathbf{U}$ encodes the user u_j to the real-valued embedding $\mathbf{u}_j$, which is more informative. In the same way, we respectively create an item embedding matrix $\mathbf{I} \in \mathbb{R}^{|\mathcal{I}| \times d}$ and item attribute embedding matrices, e.g. item category embedding matrix $\mathbf{C} \in \mathbb{R}^{|\mathcal{C}| \times d}$.

Positional Embedding: As we mentioned above, traditional HIN based recommendation methods usually neglect the sequential pattern of the interactions. Motivated by the recent works of transformer [20, 2], we correlate each item with a learnable position embedding $\mathbf{P} \in \mathbb{R}^{|\mathcal{P}| \times d}$ to capture the sequential pattern of the items.

$$\hat{\mathbf{I}}_u = \begin{bmatrix} \mathbf{i}_t + \mathbf{p}_1 \\ \mathbf{i}_{t-1} + \mathbf{p}_2 \\ \cdots \\ \mathbf{i}_1 + \mathbf{p}_t \end{bmatrix}, \tag{3.1}$$

where $S_u = \{i_1, \cdots, i_{t-1}, i_t\}$ [1] is the interacted item sequence for a specific user u, sorted by the time t. Note that we add the position embedding in a reverse order to capture the relevant distance to the target item.

[1] We truncate or padding the sequence with a max length of 200.

3.2 Sequence-aware Heterogeneous Message Passing Layer To capture high-order heterogeneous collaborative information and the sequential information, we first construct an HIN enriching user-item interactions with added item attributes as shown in Fig. 1. We also model a user's interacted item sequence with a sequence-aware attention mechanism, in order to capture the user's dynamic interests. In this way, we not only can better model user preferences but also alleviate the sparsity of the interactions. In the following, we will first present a single graph convolution layer to model item embedding and user embedding with considering heterogeneous information and sequential information, and then generalize it to multiple layers.

3.2.1 Item Modeling with Heterogeneous Information To alleviate the sparsity problem, we add item attribute information to the user-item bipartite graph and create an HIN that includes different types of nodes. Inspired by HGAT [8], in this part, we present our proposed message passing layers which consider the heterogeneous information.

Taking the item node as an example, it has different types of neighboring nodes such as users, categories, cities, etc. On one hand, different types of neighboring nodes may have different impacts on it. For example, the category of one item may be more informative than the user who interacted with it. On the other hand, different neighboring nodes of the same type could also have different importance. For example, different users may have different preference on one item. To capture both the different importance at both node level and type level, we design a dual-level attention mechanism when aggregating embeddings from neighboring nodes.

Node-level Attention: We design the node-level attention to learn the different importance of neighboring nodes with the same type and aggregate the representation of these neighbors to form a specific type embedding. Formally, given a specific item v and its neighboring node $v' \in \mathcal{N}_v^{\tau'}$ with type τ', the weight coefficient $\alpha_{vv'}$ of the specific node pair (v, v') can be formulated as follows:

$$\alpha_{vv'} = \frac{\exp(\sigma(\mathbf{a}_{\tau'}^{\top} \cdot [\mathbf{i}_v || \mathbf{h}_{v'}]))}{\sum_{k \in \mathcal{N}_v^{\tau'}} \exp(\sigma(\mathbf{a}_{\tau'}^{\top} \cdot [\mathbf{i}_v || \mathbf{h}_k]))}, \tag{3.2}$$

where $\sigma(\cdot)$ is the activation function, such like LeakReLU, $\mathbf{a}_{\tau'}$ is the attention vector for the type τ' and $||$ denotes the concatenate operation.

Then, for the item v, we can get the specific type embedding $\mathbf{g}_v^{\tau'}$ by aggregating neighboring nodes of the same type with corresponding coefficients as follows:

$$\mathbf{g}_v^{\tau'} = \sigma(\sum_{v' \in \mathcal{N}_v^{\tau'}} \alpha_{vv'} \cdot \mathbf{h}_{v'}). \tag{3.3}$$

Type-level Attention: For any type τ' belonging to the item v's neighboring node type set $\mathcal{T}$, we can get a type specific embedding $\mathbf{h}_v^{\tau'}$ following the Eq. (3.3). To capture different importance of different node types, we design a type-level attention defined as follows:

$$m_v^{\tau'} = V \cdot \tanh(\mathbf{w} \cdot \mathbf{g}_v^{\tau'} + b), \tag{3.4}$$

$$\beta_v^{\tau'} = \frac{\exp(m_v^{\tau'})}{\sum_{\tau \in \mathcal{T}} \exp(m_v^{\tau})}. \tag{3.5}$$

With the learned weights as coefficients, we can fuse these type embeddings $\mathbf{g}_v^{\tau'}$ to obtain the final item embedding $\tilde{\mathbf{i}}_v$ as follows:

$$\tilde{\mathbf{i}}_v = \sigma(\sum_{\tau' \in \mathcal{T}} \beta_v^{\tau'} \cdot \mathbf{g}_v^{\tau'}). \tag{3.6}$$

Note that the above is an example of how to obtain item embedding with heterogeneous information, other typed nodes in HIN such as attribute nodes can be modeled in the same way.

3.2.2 User Modeling with Fine-grained Static and Dynamic Interest

A great challenge for recommendation is how to accurately model user preferences. For traditional collaborative filtering or HIN based recommendation methods, on one hand, they usually view an item as an entirety, which ignore the fact that users may have different preferences on different aspects of an item; on the other hand, they always neglect the sequential information of a user's interaction history, thus failing to capture the user's dynamic interests. Therefore, for user nodes, we present a carefully designed message passing layer to capture a user's fine-grained static interests and dynamic interests. More specifically, we propose an element-wise attention mechanism that assumes each dimension of the item embedding reflects a distinct aspect of the item. In addition, to capture a user's dynamic interests, we adopt a sequence-aware self-attention mechanism where each item embedding is correlated with a position embedding, and self-attention is applied to pay attention to important items.

Element-wise Attention: Here we present the details of element-wise attention to capture user's fine-grained static preference. For a specific item i_j in user u's interaction sequence $\mathcal{S}_u$, we can calculate a weight vector $\boldsymbol{\gamma}_j$ for different aspects of item i_j as follows:

$$\boldsymbol{\gamma}_j = \tanh(\mathbf{W}_u \cdot \mathbf{i}_j + b), \tag{3.7}$$

where $\mathbf{W}_u \in \mathbb{R}^{d \times d}$, $\boldsymbol{\gamma}_j$ is the attention coefficients of different aspects, and a large γ_j^k means that the k^{th} aspect of item embedding $\mathbf{i}_j$ is strongly relevant to the user's preference.

Then we aggregate with element-wise product between the weight coefficients $\boldsymbol{\gamma}_j$ and the user integrated item $\mathbf{i}_j$ to capture the user's fine-grained static interests:

$$\mathbf{u}_s = \sum_{j \in \mathcal{S}_u} \boldsymbol{\gamma}_j \odot \mathbf{i}_j. \tag{3.8}$$

Sequence-aware Self-attention: Motivated by the self-attention mechanism widely used in NLP tasks such like machine translation[20, 2], we adopt a sequence-aware self-attention mechanism where each item $\hat{\mathbf{I}}_u$ is integrated with its position embedding and self-attention is used to pay attention to critical items, to capture user's dynamic interests over time:

$$\textsc{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}}) \cdot \mathbf{V}, \tag{3.9}$$

$$\mathbf{u}_d = \big\|_{h=1}^{H} \textsc{Attention}(\hat{\mathbf{I}}_u \mathbf{W}^Q, \hat{\mathbf{I}}_u \mathbf{W}^K, \hat{\mathbf{I}}_u \mathbf{W}^V), \tag{3.10}$$

where $\textsc{Attention}()$ calculates a weighted sum of all values, and the scale factor $\sqrt{d}$ is to avoid overly large values of the inner product result. $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ is the projection matrices and we extend the self-attention to multi-head attention by repeating it for H times and concatenate the learned embeddings to get the final user dynamic interest representation.

After getting the static interest embedding $\mathbf{u}_s$ and dynamic interest embedding $\mathbf{u}_d$, we combine them with a balance weight to get the final user embedding $\tilde{\mathbf{u}}$:

$$\tilde{\mathbf{u}} = \lambda \mathbf{u}_d + (1 - \lambda) \mathbf{u}_s \tag{3.11}$$

3.2.3 High-order Propagation The above shows a single messaging passing layer with heterogeneous information and sequential information, which aggregates information from the first-order neighbors. To capture the high-order collaborative information, we can stack it to multiple layers in which each layer takes the last layer's output representation as its input [2]. After L-layer embedding propagation, we can get output embeddings of L different layers.

3.3 Prediction and Optimization The embeddings of different layers may have different contributions in reflecting user preferences, following [23], we concatenate the representation of each layer to constitute the final embedding for both users and items:

$$\mathbf{u} = \mathbf{u}^1 \,||\, \mathbf{u}^2 \,||\, \cdots \,||\, \mathbf{u}^L, \quad \mathbf{i} = \mathbf{i}^1 \,||\, \mathbf{i}^2 \,||\, \cdots \,||\, \mathbf{i}^L. \tag{3.12}$$

Finally, we use the simple dot product to estimate the user's preference towards the target item:

$$\hat{y}(u,i) = \mathbf{u}^\top \mathbf{i}. \tag{3.13}$$

To optimize our model, we use the Bayesian Personalized Ranking (BPR) loss[15] as our loss function:

$$\mathcal{L} = \sum_{i \in S_u, j \notin S_u} -\ln \sigma(\hat{\mathrm{y}}_{\mathrm{ui}} - \hat{\mathrm{y}}_{\mathrm{uj}}) + \eta\,||\Theta||, \tag{3.14}$$

where $\sigma(\cdot)$ is the sigmid function, Θ denotes all the trainable parameters and η is the regularization coefficient, S_u is the interaction sequence of the user u, and for each positive sample (u,i), we sample a negative sample j that the user not interacted for training.

4 Experiments

4.1 Experimental Setup

Datasets. To verify the effectiveness of our method, we conduct extensive experiments on three real-world datasets. Movielens is a widely used benchmark dataset for recommendation task.In our experiments, we adopt a small version of 100K interactions and a larger version of 1M interactions. Yelp is a local business recommendation dataset which records the user ratings on local businesses. The details of them are shown in Table 1.

Evaluation Metrics. In our experiments, we use the leave-one-out strategy for evaluation. For each user, the last interacted item is selected out for testing, and the remaining data for training. Following previous works [5, 4], in test set, we randomly select 99 items that are not interacted by the user as negative samples and rank the 100 sampled items from high scores to low scores. For a fair comparison with the baseline methods, we use the same negative sample set for each (user, item) pair in the test set for all the methods. We evaluate the recommendation performance with two popular metrics Hit Ratio (HR) and the Normalized Discounted Cumulative Gain (NDCG).

Table 1: Details of the Datasets

Dataset (Density)	#Users	#Items	#Categories	#Cities	#Average inteactions /user
ML100K (9.036%)	943	1,152	19	-	103.9
ML1M (5.057%)	6,040	3,260	18	-	165.3
Yelp (0.097%)	10,002	10,373	492	58	10.0

Baselines. We compare our method with three groups of recommendation baseline methods: collaborative filtering methods(MF-BPR, NeuMF, NGCF), HIN-based recommendation methods(NeuACF, HeRec) , and sequential recommendation methods(NARM, SR-GNN).

Implementation Details. For all the methods, we apply a grid search for hyperparameters. For NGCF and SR-GNN, the layer of GNN is searched from 1 to 4. We implement our proposed model based on Tensorflow. The dimension of embeddings d is set as 64. For the self-attention network, the attention head number H is set as 8. The hyperparameter λ to balance the weights of a user's dynamic interests and static interests is set as 0.5 and 0.2 for MovieLens and Yelp, respectively. In addition, the learning rate is 0.0005 for MovieLens and 0.00005 for Yelp. The coefficient of L2 normalization η for all the datasets is set to 10^{-5}. We set the depth of our proposed SHCF L as 4. We randomly initialize the model parameters with Xavier initializer, then use the Adam as the optimizer. To avoid over-fitting, we apply early stopping strategy and apply dropout (dropout rate is 0.1) in every layer of our proposed SHCF.

4.2 Comparison of Performance We first compare the recommendation performance of all the methods. For a fair comparison, the embedding dimension of all the methods is set as 64. Table 2 shows the experiment results of different methods. We have the following observations: 1) HIN based recommendation methods generally perform better than traditional collaborative filtering methods. They especially have great improvements on the sparse dataset (i.e., Yelp), which illustrates that applying HIN to incorporate side information for recommendation can alleviate the data sparsity problem and improve recommendation performance. 2) For

[2]According to our experiments, fine-grained and dynamic user interest modeling layer is taken as the L-th layer for user modeling.

Table 2: Recommendation performance of different models. The best result in each row is bold and the second best result is underlined. The improvements of our method over the second best models are shown in the last column.

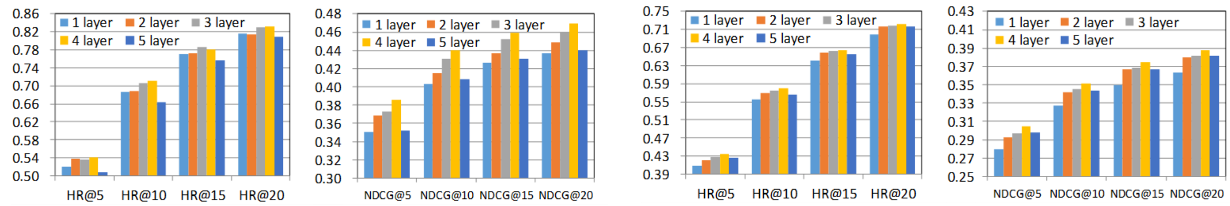
Dataset	Metrics	BPR-MF	NeuMF	NGCF	NeuACF	HeRec	NARM	SR-GNN	SHCF	Improve
ML100K	HR@5	0.4030	0.4057	0.4274	0.4337	0.4255	0.5228	0.5010	**0.5414**	3.56%
	NDCG@5	0.2747	0.2676	0.2889	0.2874	0.2798	0.3659	0.3510	**0.3859**	5.47%
	HR@10	0.5801	0.5689	0.5864	0.6034	0.6012	0.6723	0.6660	**0.7108**	5.73%
	NDCG@10	0.3312	0.3127	0.3402	0.3420	0.3325	0.4142	0.4048	**0.4401**	6.25%
	HR@15	0.6787	0.6706	0.6649	0.7084	0.6981	0.7529	0.7598	**0.7817**	2.88%
	NDCG@15	0.3573	0.3462	0.3611	0.3697	0.3521	0.4354	0.4298	**0.4592**	5.47%
	HR@20	0.7455	0.7595	0.7434	0.7720	0.7524	0.8038	0.8048	**0.8324**	3.43%
	NDCG@20	0.3731	0.3672	0.3796	0.3847	0.3721	0.4475	0.4396	**0.4693**	4.87%
ML1M	HR@5	0.4921	0.5092	0.5017	0.5050	0.4923	0.6713	0.6634	**0.6927**	3.18%
	NDCG@5	0.3376	0.3511	0.3437	0.3508	0.3455	0.5201	0.5233	**0.5299**	1.26%
	HR@10	0.6577	0.6803	0.6688	0.6684	0.6601	0.7603	0.7699	**0.7964**	3.19%
	NDCG@10	0.3910	0.4066	0.3977	0.4038	0.3982	0.5565	0.5580	**0.5639**	1.06%
	HR@15	0.7551	0.7761	0.7587	0.7593	0.7403	0.8230	0.8184	**0.8503**	3.32%
	NDCG@15	0.4168	0.4320	0.4216	0.4279	0.4194	0.5687	0.5709	**0.5785**	1.33%
	HR@20	0.8159	0.8369	0.8167	0.8232	0.8105	0.8602	0.8584	**0.8844**	2.81%
	NDCG@20	0.4311	0.4463	0.4353	0.4430	0.4328	0.5723	0.5803	**0.5968**	2.84%
Yelp	HR@5	0.3077	0.3571	0.4097	0.4094	0.3982	0.3490	0.3754	**0.4421**	7.91%
	NDCG@5	0.2086	0.2419	0.2855	0.2844	0.2765	0.2373	0.2565	**0.3100**	8.58%
	HR@10	0.4325	0.5018	0.5584	0.5553	0.5505	0.4900	0.5208	**0.5878**	5.27%
	NDCG@10	0.2488	0.2885	0.3335	0.3314	0.3311	0.2828	0.3035	**0.3572**	7.11%
	HR@15	0.5084	0.6006	0.6434	0.6504	0.6423	0.5851	0.6054	**0.6725**	3.40%
	NDCG@15	0.2689	0.3146	0.3544	0.3565	0.3499	0.3080	0.3259	**0.3796**	6.48%
	HR@20	0.5643	0.6717	0.7005	0.7138	0.6923	0.6496	0.6684	**0.7260**	1.71%
	NDCG@20	0.2821	0.3315	0.3686	0.3715	0.3603	0.3232	0.3408	**0.3922**	5.57%

the dense datasets ML100K and ML1M that users have adequate interaction behaviors (the average number of interactions per user is 103.9 and 165.3 respectively), sequential recommendation methods perform better than collaborative filtering methods and HIN based recommendation methods. But for the sparse dataset Yelp where the average number of interactions per user decreases to 10.0, the performance of sequential recommendation methods significantly decline. It illustrates the limitation of sequential recommendation methods in modeling user embeddings based on only the user's own sequential item interactions without considering the collaborative information of similar users or items when the user does not have sufficient interaction records. 3) Our proposed model SHCF consistently outperforms all the baselines on all the datasets including the two dense datasets (i.e., ML100K and ML1M) and one sparse dataset (i.e., Yelp). These results verify the effectiveness of SHCF in modeling users and items in both sparse and dense datasets by fully considering the high-order heterogeneous collaborative information and sequential information.

Table 3: Comparison of SHCF Variants

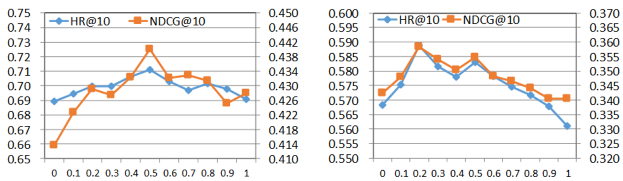
Dataset	Metrics	SHCF w/o DS	SHCF -S	SHCF -D	SHCF
ML100K	HR@5	0.3881	0.5306	0.5350	**0.5414**
	NDCG@5	0.2570	0.3645	0.3779	**0.3859**
	HR@10	0.5619	0.6892	0.6908	**0.7108**
	NDCG@10	0.3121	0.4136	0.4280	**0.4401**
	HR@15	0.6892	0.7636	0.7692	**0.7817**
	NDCG@15	0.3459	0.4388	0.4521	**0.4592**
	HR@20	0.7455	0.8107	0.8098	**0.8324**
	NDCG@20	0.3598	0.4473	0.4609	**0.4693**
Yelp	HR@5	0.4241	0.4355	0.4273	**0.4421**
	NDCG@5	0.2997	0.3049	0.3014	**0.3100**
	HR@10	0.5572	0.5767	0.5694	**0.5878**
	NDCG@10	0.3397	0.3479	0.3461	**0.3572**
	HR@15	0.6385	0.6572	0.6543	**0.6725**
	NDCG@15	0.3612	0.3714	0.3685	**0.3796**
	HR@20	0.6986	0.7139	0.7090	**0.7260**
	NDCG@20	0.3754	0.3847	0.3813	**0.3922**

4.3 Effect of Different Components Since there are many components in our model, we also analyze their impacts via an ablation study. Table 3 shows the performance of our model and its three variants on ML100K and Yelp. The basic model SHCF w/o DS learns the user embedding in the same way as other types of nodes with L message passing layers based on the HIN. It does not consider the user's fine-



(a) ML100K (b) Yelp

Figure 3: Effect of the number of message passing layers.



(a) ML100K (b) Yelp

Figure 4: Effect of the balance coefficient λ.

grained static interest modeling (i.e., without element-wise attention) and dynamic interests (i.e., without sequence-aware self-attention). SHCF-D considers the user's dynamic interests while ignores the fine-grained static interest modeling. SHCF-S only consider the user's fine-grained static interests.

We can see from Table 3 that SHCF-DS which removes both fine-grained static interest and dynamic interest modeling performs worst among all the models on all the datasets. SHCF-D and SHCF-S improve SHCF w/o DS, demonstrating the effectiveness of considering the user's dynamic interests or fine-grained static interests. Finally, SHCF significantly outperforms all the variants by considering both the user's dynamic interests and fine-grained static interests.

4.4 Parameter Analysis of SHCF In this section, we investigate the sensitivity of some important parameters of our model on dataset ML100K and Yelp.

Effect of the number of message passing layers. We first test the impact of different number of message passing layers on the recommendation performance. The results are shown in Fig. 3. As we can see, on both datasets, our model performs best with 4 layers. We think the reason may be that when SHCF has a less number of layers, it could not capture the higher-order collaborative relationships, and when the layer number gets larger, it may cause the oversmoothing problem and bring massive noise to the model, thus decreasing the recommendation performance.

Effect of the balance coefficient λ. As presented in Eq. (3.11), we model user embedding by combining his dynamic interests and static interests with a balance coefficient λ. The higher the value of λ, the more attention we paid to the user's dynamic interests. In this part, we study the performance with different values of λ from 0 to 1. As shown in Fig. 4, for both of the datasets, the performance of our model first rises with the growth of λ and then drops, which illustrates that a balance of the user's dynamic interests and static interests is important to the recommendation performance. On ML100K, our model achieves the best performance when λ is 0.5, which views the user's dynamic interests as equally important as static interests. While on Yelp, due to the short range of user's interaction sequence, a smaller λ (0.2) achieves the best performance, which shows the dynamic interests contribute less than the static interests.

5 Conclusion

In this paper, we propose a novel sequence-aware heterogeneous graph neural collaborative filtering model for recommendation, which takes full advantage of both the sequential pattern of user-item interactions and high-order heterogeneous collaborative signals. Particularly, we first construct an HIN that enriches user-item interactions with additional item attributes. And then novel message passing layers are designed for learning user and item embeddings on the HIN. For user embedding, we capture a user's fine-grained preferences on different aspects of an item with an element-wise attention mechanism. We also consider the user's dynamic interests over time by aggregating the item interaction sequence with a sequence-aware self-attention mechanism. For item embedding, when aggregating its neighboring information, we consider the importance of different neighboring nodes of different types via dual-level attention. Extensive experimental results demonstrated that our proposed model consistently outperforms the state-of-the-art methods across either dense datasets or sparse datasets.

6 Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. U20B2045, 61772082, 61702296, 62002029, 61806020). Chen Li is also supported by 2020 Tencent Rhino-Bird Elite Training Program.

References

[1] R. v. d. Berg, T. N. Kipf, and M. Welling, *Graph convolutional matrix completion*, KDD, (2018).

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[3] W. Feng and J. Wang, *Incorporating heterogeneous information for personalized tag recommendation in social tagging systems*, in SIGKDD, 2012, pp. 1276–1284.

[4] X. Han, C. Shi, S. Wang, S. Y. Philip, and L. Song, *Aspect-level deep collaborative filtering via heterogeneous information networks.*, in IJCAI, 2018, pp. 3393–3399.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, *Neural collaborative filtering*, in WWW, 2017, pp. 173–182.

[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, *Session-based recommendations with recurrent neural networks*, arXiv preprint arXiv:1511.06939, (2015).

[7] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, *Parallel recurrent neural network architectures for feature-rich session-based recommendations*, in RecSys, 2016, pp. 241–248.

[8] L. Hu, T. Yang, C. Shi, H. Ji, and X. Li, *Heterogeneous graph attention networks for semi-supervised short text classification*, in EMNLP, 2019, pp. 4823–4832.

[9] W.-C. Kang and J. McAuley, *Self-attentive sequential recommendation*, in ICDM, IEEE, 2018, pp. 197–206.

[10] Y. Koren, *Factorization meets the neighborhood: a multifaceted collaborative filtering model*, in SIGKDD, 2008, pp. 426–434.

[11] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.

[12] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, *Neural attentive session-based recommendation*, in CIKM, 2017, pp. 1419–1428.

[13] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, *Stamp: short-term attention/memory priority model for session-based recommendation*, in SIGKDD, 2018, pp. 1831–1839.

[14] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, *Personalizing session-based recommendations with hierarchical recurrent neural networks*, in RecSys, 2017, pp. 130–137.

[15] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, *Bpr: Bayesian personalized ranking from implicit feedback*, UAI, (2009).

[16] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, *Factorizing personalized markov chains for next-basket recommendation*, in WWW, 2010, pp. 811–820.

[17] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, *Heterogeneous information network embedding for recommendation*, IEEE TKDE, 31 (2018), pp. 357–370.

[18] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, *A survey of heterogeneous information network analysis*, IEEE TKDE, 29 (2016), pp. 17–37.

[19] X. Su and T. M. Khoshgoftaar, *A survey of collaborative filtering techniques*, Advances in artificial intelligence, 2009 (2009).

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, in NeurIPS, 2017, pp. 5998–6008.

[21] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, *Kgat: Knowledge graph attention network for recommendation*, in SIGKDD, 2019, pp. 950–958.

[22] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, *Neural graph collaborative filtering*, in SIGIR, 2019, pp. 165–174.

[23] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, *Heterogeneous graph attention network*, in WWW, 2019, pp. 2022–2032.

[24] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, *Social recommendation with strong and weak ties*, in CIKM, 2016, pp. 5–14.

[25] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, *Session-based recommendation with graph neural networks*, in AAAI, vol. 33, 2019, pp. 346–353.

[26] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, *Deep matrix factorization models for recommender systems.*, in IJCAI, 2017, pp. 3203–3209.

[27] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai, *Hop-rec: high-order proximity for implicit recommendation*, in RecSys, 2018, pp. 140–144.

[28] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, *Personalized entity recommendation: A heterogeneous information network approach*, in SIGKDD, 2014, pp. 283–292.

[29] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, *Meta-graph based recommendation fusion over heterogeneous information networks*, in SIGKDD, 2017, pp. 635–644.

[30] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, and X. Li, *Connecting social media to e-commerce: Cold-start product recommendation using microblogging information*, IEEE TKDE, 28 (2015), pp. 1147–1159.